4-D/RCS: A Reference Model Architecture for Demo III*

James S. Albus
Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology, Gaithersburg, MD 20899

ABSTRACT

4-D/RCS is a reference model architecture that integrates the NIST (National Institute of Standards and Technology) RCS Real-time Control System [Albus & Meystel],with the German (Universitat der Bundeswehr Munchen) VaMoRs 4-D approach to dynamic machine vision [Dickmanns]. 4-D/RCS will provide the DOD Demo III project with an open system operational architecture that will facilitate the integration of a wide variety of subsystems. These include: (1) a foveal/peripheral CCD color camera pair, (2) a Laser Range Imager, (3) a MAPS inertial guidance package, (4) a GPS satellite navigation system, (5) stereo image processing algorithms developed at the NASA Jet Propulsion Lab, and (6) a HMMWV telerobotic driving system with interfaces for a wide variety of mission packages.

I. INTRODUCTION

4-D/RCS is a reference model architecture for the design and development of intelligent vehicle systems and software, and to provide the theoretical basis for future standards.

Df. 1. reference model architecture
an architecture in which the entire collection of entities, relationships, and information units involved in interactions between and within subsystems are defined and modeled

Df. 2. intelligent system
a system with the ability to act appropriately in an uncertain environment

Df. 3. appropriate action
that which increases the probability of success

Df. 4. success
the achievement of mission goals

Df. 5. mission goal
a desired result that a mission is designed to achieve or maintain

A desired result may be a state or some integral measure of a state-time history, such as average value, variance, limit on deviation or expenditure of energy or time, or limit on time of arrival.

Df. 6. mission
the highest level task assigned to the system under consideration

In the case of a scout platoon attached to a battalion, the mission is the task given to the platoon by battalion headquarters.

II. FUNCTIONAL ELEMENTS

Df. 7. functional elements
the fundamental computational processes from which the system is composed

Axiom 1. The functional elements of an intelligent system are sensory processing, world modeling, value judgment, and behavior generation.

Df. 8. sensory processing
sensory processing is a set of processes by which sensory data interacts with prior knowledge to detect or recognize useful information about the world

Sensory processing accepts signals from sensors that measure properties of the external world or conditions internal to the system itself. Sensory processing scales, windows, and filters data, computes observed features and attributes, and compares them with predictions from internal models. Correlations between sensed observations and internally generated expectations are used to detect events and recognize entities and situations. Differences between

* This paper is a condensation of 4-D/RCS: A Reference Model Architecture for Demo III, NISTIR 5994, March 1997

sensed observations and internally generated predictions are used to update internal models. Sensory processing also computes attributes of entities and events, and clusters, or groups, recognized entities and detected events into higher order entities and events.

Df. 9. world modeling
world modeling is a set of proceses that construct and maintain a world model that can be used for feedback control as well as for the generation of predictions of sensory signals and simulation of plans.

World modeling performs four principal functions:

1. It provides a best estimate of the state of the world to be used as the basis for predicting sensory feedback and planning future actions.
2. It maintains a knowledge database about objects, agents, situations, and relationships, including knowledge of task skills and laws of nature.
3. It predicts (possibly with several hypotheses) sensory observations based on the estimated state of the world. Predicted signals can be used by sensory processing to configure filters, masks, windows, and templates for correlation, model matching, and recursive estimation.
4. It simulates results of possible future plans based on the estimated state of the world and planned actions. Simulated results are evaluated by the value judgment system in order to select the best plan for execution.

Df. 10. world model
an internal representation of the world

The world model may include models of portions of the environment, as well as models of objects and agents, and a system model that includes the intelligent system itself. The world model is stored in a dynamic distributed knowledge database that is maintained by the world modeling process.

Df. 11. knowledge database
the data structures and the static and dynamic information that collectively form the intelligent system's world model

The knowledge database stores information about the world in the form of structural and dynamic models, state variables, attributes and values, entities and events, rules and equations, task knowledge, images, and maps.

The knowledge database has three parts:

a) A long term memory containing symbolic and iconic representations of all the generic and specific objects, events, and rules that are known to the intelligent system.

b) A short term memory containing iconic and symbolic representations of geometric entities and events that are the subject of current attention.

c) Internal representation of immediate experience consisting of sensor signals and current values of observed, estimated, and predicted attributes.

Df. 12. value judgment
value judgment is a process that:

a) computes cost, risk, and benefit of actions and plans,

b) estimates the importance and value of objects, events, and situations,

c) assesses the reliability of information,

d) calculates the rewarding or punishing effects of perceived states and events.

Value judgment evaluates perceived and planned situations thereby enabling behavior generation to select goals and set priorities. It computes what is important (for attention), and what is rewarding or punishing (for learning). It assigns values to recognized objects and events, and computes confidence factors for observed, estimated, and predicted attributes and states.

Df. 13. behavior generation
behavior generation is the planning and control of actions designed to achieve behavioral goals.

Behavior generation accepts task commands with goals and priorities, formulates and/or selects plans, and controls action. Behavior generation develops or selects plans by using a priori task knowledge and value judgment functions combined with real-time information provided by world modeling to find the best assignment of tools and resources to agents, and to find the best schedule of actions (i.e., the most efficient plan to get from an anticipated starting state to a goal state). Behavior generation controls action by both feedforward actions and by feedback error compensation.

Axiom 2. The functional elements and knowledge database of an intelligent system can be distributed over a set of computational nodes, and be represented within each computational node by a set of processes interconnected by a communication system that transfers information between them.

III. COMPUTATIONAL NODES

Df. 14. node
a part of a control system that processes sensory information, maintains the generic and dynamic world model (including possibly several hypotheses) computes values, and plans and executes tasks

A typical 4-D/RCS node contains a Behavior Generation (BG) process, a World Modeling (WM) process, a Sensory Processing (SP) process, a Value Judgment (VJ) process, and a Knowledge Database (KD). All the processes in an 4-D/RCS node may have input and output connections to an Operator Interface.

Figure 1 illustrates the relationships in a single node of the 4-D/RCS architecture. The interconnections between SP, WM, and BG processes close a reactive feedback control loop between sensory measurements and commanded action. The interconnections between BG, WM, and VJ processes enable deliberative planning and reasoning about future actions. The interconnections between SP, WM, and VJ processes enable knowledge acquisition, situation evaluation, and learning.

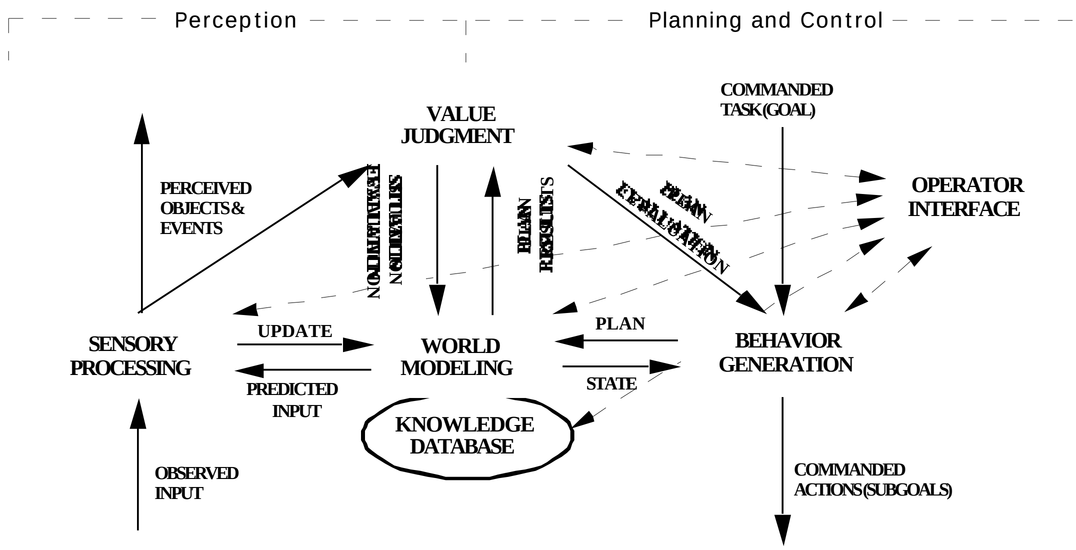

Figure 1. A node in the 4-D/RCS reference model architecture. The functional elements of an intelligent systme are behavior generation (planning and control), sensory processing (filtering, detection, recognition, grouping), world modeling (store and retrieve knowledge and predict future states), and value judgment (compute cost, benefit, risk, importance, and uncertainty). These are supported by a knowledge database and a communication system that interconnects the functional processes and the knowledge database. This collection of processes and their interconnections make up a generic node in the 4-D/RCS reference model architecture. Each process in the node may have an operator interface.

The connections to the Operator Interface enable a human supervisor to input commands, to override or modify system behavior, to perform various types of teleoperation, to switch control modes (e.g., automatic, teleoperation, single step, pause), and to observe the values of state variables, images, maps, and entity attributes. The Operator Interface can also be used for maintenance, programming, and debugging.

The generic node illustrated in Figure 1 can be used to construct a distributed hierarchical reference model architecture. Each node in the 4-D/RCS architecture corresponds to a functional unit in a military command and control hierarchy. Depending on where the generic node resides in the hierarchy, it might serve as an intelligent controller for an actuator, a subsystem, a vehicle, a section, a platoon, a company, battalion, or higher level organizational unit. Each generic node (or a set of processes within a node) might be implemented as an intelligent supervised-autonomy controller, or as a human person or management unit, at any level in the military command and control structure.

Axiom 3. The complexity inherent in intelligent systems can be managed through hierarchical layering.

Intelligent systems are inherently complex. Hierarchical layering is a common method for organizing complex systems that has been used in many different types of organizations throughout history for effectiveness and efficiency of command and control. In a hierarchical control system, higher level nodes have broader scope and longer time horizons with less concern for detail. Lower level nodes have narrower scope and shorter time horizons with more focus on detail. At no level does a node have to cope with both broad scope and high level of detail. This enables the design of systems of arbitrary complexity, without computational overload in any node or at any level.

Hierarchical layering makes the 4-D/RCS architecture extensible and portable. Generic nodes can be designed and customized for use at many different levels by embedding different functional algorithms and knowledge. In the 4-D/RCS reference architecture, behavior generation processes in nodes at the upper levels in the hierarchy make long range strategic plans consisting of major milestones, while lower level behavior generation processes successively decompose the long range plans into short range tactical plans with detailed activity goals. Sensory processing processes at lower levels process data over local neighborhoods and short time intervals, while at higher levels, they integrate data over long time intervals and large spatial regions. At low levels, the knowledge database is short term and fine grained, while at higher levels it is broad in scope and generalized. At every level, feedback loops are closed to provide reactive behavior, with high-bandwidth fast-response loops at lower levels, and slower more deliberative reactions at higher levels.

At each level, state variables, entities, events, and maps are maintained to the resolution in space and time that is appropriate to that level. At each successively lower level in the hierarchy, as detail is geometrically increased, the range of computation is geometrically decreased. Also, as temporal resolution is increased, the span of interest decreases. This produces a ratio that remains relative constant throughout the hierarchy. As a result, at each level, behavior generation functions make plans of roughly the same number of steps. At higher levels, the space of planning options is larger and world modeling simulations are more complex, but there is more time available between replanning intervals for planning processes to search for an acceptable or optimal plan. Thus, hierarchical layering keeps the amount of computing resources needed for behavior generation in each node within manageable limits.

Also at each level, lower level entities are grouped into higher level entities. The effect is to geometrically increase the scope and encapsulate the detail of entities and events observed in the world. Thus, at each level, sensory processing functions compute entities that contain roughly the same number of sub entities.

Along the time line from the present (t = 0), short term memory is much more detailed than long-term memory, and plans for the immediate future are much more detailed than plans for the long term.

This kind of layering is typical of the military command and control hierarchy. At the top, strategic objectives are chosen and priorities defined that influence the selection of goals and the prioritization of tasks throughout the entire hierarchy. The details of execution are left to subordinates.

At intermediate levels, tasks with goals and priorities are received from the level above, and sub tasks with sub goals and attention priorities are output to the level below. In the intelligent vehicle environment, intermediate level tasks might be of the form <go to position at map coordinates x,y>, <advance in formation along line z>, <engage enemy units at time t>, etc. The details of execution are left to subordinates.

At each level in the 4-D/RCS hierarchy, higher level more global tasks are decomposed and focused into concurrent strings of more narrow and finer resolution tasks. The effect of each hierarchical level is thus to geometrically refine the detail of the task and limit the view of the world, so as to keep computational loads within limits that can be handled by individual intelligent agents, such as 4-D/RCS nodes or ordinary human beings.

Axiom 3. The complexity of the real world environment can be managed through focusing attention.

Intelligent systems must operate in a real world environment which is infinitely rich with detail. The real world environment contains a practically infinite variety of real objects, such as the ground, rocks, grass, sand, mud, trees, bushes, buildings, posts, ravines, rivers, roads, enemy and friendly positions, vehicles, weapons, and people. The environment also contains elements of nature, such as wind, rain, snow, sunlight, and darkness. All of these objects and elements have states and may cause, or be part of, events and situations. The environment also contains a practically infinite regression of detail, and the world itself extends indefinitely far in every direction.

Yet, the computational resources available to any intelligent system are finite. No matter how fast and powerful computers become, the amount of computational resources that can be embedded in any practical system will be limited. Therefore, it is imperative that the intelligent system focus the available computing resources on what is important, and ignore what is irrelevant.

Fortunately, at any point in time and space, most of the detail in the environment is irrelevant to the immediate task of the intelligent system. Therefore, the key to building practical intelligent systems lies in understanding how to focus the available computing resources on what is important and ignore what is irrelevant. The problem of distinguishing what is important from what is irrelevant must be addressed from two perspectives: top down and bottom up.

Top down, what is important is defined by behavioral goals. The intelligent system is driven by high-level goals and priorities to focus attention on objects specified by the task, using resources identified by task knowledge as necessary for successfully accomplishing given goals. Top down goals and high-level perceptions generate expectations of what objects and events might be encountered during the evolution of the task and which are important to achieving the goal.

Bottom up, what is important is the unexpected, unexplained, unusual, or out-of-limits. At each level, sensory processing functions detect errors between what is expected and what is observed. Error signals are processed at lower levels first. Control laws in lower level behavior generation processes generate corrective actions designed to correct the errors and bring the process back to the plan. However, if low level reactive control laws are incapable of correcting the differences between expectations and observations, errors filter up to higher levels where plans may be revised and goals restructured. The lower levels are thus the first to compute attributes of signals or images that indicate problems or emergency conditions, such as limits being exceeded on position, velocity, acceleration, vibration, pressure, force, current, voltage, or temperature. The lower levels of control are also the first to act to correct, or compensate for errors.

In either case, hierarchical layering provides a mechanism for focusing the computational resources of the lower levels on particular regions of time and space. Higher level nodes with broad perspective and long planning horizon determine what is important, while the lower levels detect anomalies and attend to details of correcting errors and following plans. In each node at each level, computing resources are focused on issues relevant to the decisions that must be made within the scope of control and time horizon of that node.

The region in space and time that is most relevant to the behavioral choices of an intelligent system is the region around the “here and now.” An intelligent system exists a the center of its own egosphere. The relevance of entities and events in the world are typically inversely proportional to their distance from the origin of this egosphere (i.e., here). The intelligent system also exists at the point in time labeled “now” (i.e., t = 0). The relevance of events also tends to be inversely proportional to their time from “now.”

Focusing of attention can be accomplished through masking, windowing, and filtering based on object and feature hypotheses and task goals. It can also be accomplished by pointing high resolution regions of sensors at objects-of-attention. For example in the human eye, the visual field is sampled at high resolution in the foveal region, and lower resolution in the periphery. Similarly, tactile sensors are closely spaced to produce high resolution in the finger tips, lips, and tongue with much lower resolution in other regions of the skin. The foveal area of the eyes and the high resolution tactile sensory regions of the fingers and lips are behaviorally positioned so as to apply the maximum number of sensors to objects of attention. High resolution sensors are scanned over the world to explore the regions of highest interest to the goals of the task. The result is to make the largest possible number of high resolution measurements of the most important entities and events in the environment and to ignore or relegate to the background those entities and events that are unimportant.

At each level in the 4-D/RCS sensory processing hierarchy, attention is used to mask, filter, and window sensory data and to focus computational resources on objects and events that are important to the mission goal. This keeps the computational load of processing sensory data within manageable limits at all levels of the hierarchy.

IV. THE 4-D/RCS HIERARCHY

An example of an 4-D/RCS reference model architecture for a single individual intelligent vehicle is shown in Figure 2. This diagram consists of a hierarchy of control nodes, each of which contain processes corresponding to the functional elements illustrated in Figure 1. Each node consists of a behavior generation (BG), world modeling (WM), and sensory processing (SP), and knowledge database (KD) (not shown in Figure 2). Most nodes also contain a value judgment (VJ) process (hidden behind the WM process in Figure 2). Each of the nodes can therefore function as an intelligent controller. An operator interface may access processes in all nodes at all levels.

Figure 2 illustrates a vehicle system with four subsystems: locomotion, mission package, communication, and attention. Each of the four subsystems have one or more mechanisms, each of which have one or more actuators and sensors. For example, the locomotion subsystem may consist of a navigation and driving controller with several subordinate controllers for steering, braking, throttle, and gear shift, plus ignition, lights, horn, and turn signals, each of which has one or more actuators and sensors. The communication subsystem might consist of a message encoding subsystem, a protocol syntax generator, and communications bus interface, plus antenna pointing and band selection actuators. The attention subsystem might consist of mechanisms that use cameras, LADARs, FLIRs, radar, and acoustic sensors to detect and track objects, surfaces, edges and points, and compute trajectories for laser range finders, or pan, tilt, and focus actuators. The vehicle control system should be able to incorporate a variety of modular mission packages, each of which may contain a number of sensors and actuators. For example, a weapons mission package might have loading, aiming, and firing subsystems each with a number of sensors and actuators. A reconnaissance, surveillance, and target acquisition (RSTA) mission package might contain a variety of sensors and sensor pointing actuators all of which need to be coordinated to successfully achieve behavioral goals.

The operator interface (OI) provides the capability for the operator to interact with the system at any time at a number of different levels -- to adjust parameters, to change speed, to select or verify targets, or to authorize the use of weapons. The OI provides a means to insert commands, change missions, halt the system, alter priorities, perform identification friend-or-foe (IFF), or monitor any of the system functions. The OI can send commands or requests to any BG process, or display information from any SP, WM, or VJ process. It can display any of the state variables in the KD at a rate and latency dictated by the communications bandwidth. Using the OI, a human operator can view situational maps with topographic features and both friendly and enemy forces indicated with overlays. The operator may use the OI to generate graphics images of motion paths, or display control programs (plans) in advance, or while they are being executed. The OI may also provide a mechanism to run diagnostic programs in the case of system malfunctions.

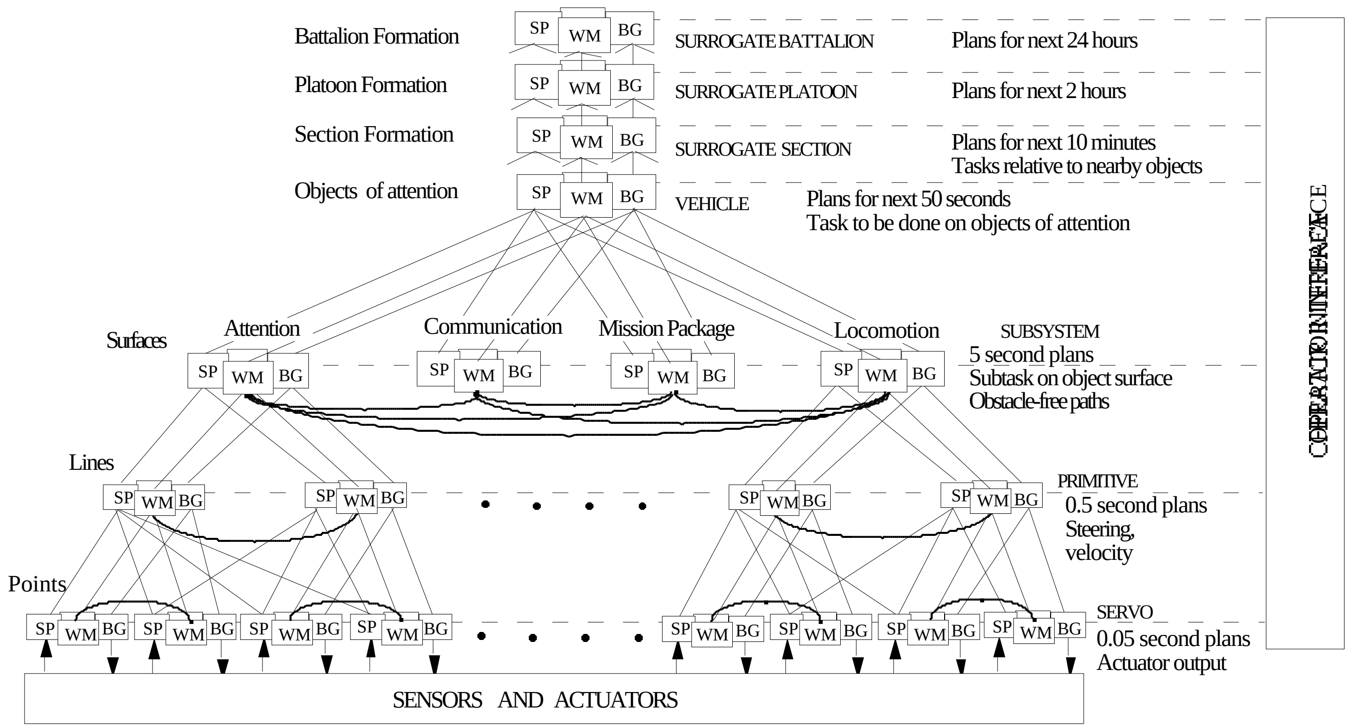

Figure 2. A 4-D/RCS reference model architecture for an individual vehicle. Processing nodes are organized such that the behavior generation (BG) processes form a command tree. Information in the knowledge database (KD) is shared between world modeling (WM) processes in nodes within the same subtree. KD structures are not shown in this figure. On the right, are examples of the functional characteristics of the behavior generation (BG) processes at each level. On the left, are examples of the type of entities recognized by the sensory processing (SP) processes and stored by the WM in the KD knowledge database at each level. Sensory data paths flowing up the hierarchy typically form a graph, not a tree. Value judgment (VJ) processes are hidden behind WM processes. A control loop may be closed at every node. An operator interface may provide input to, and output from, processes in every node.

In Figure 2, three levels of control are shown above the node representing the individual vehicle. These three additional levels represent a surrogate chain of command that, in principle, exists above the individual vehicle. However, because each vehicle is semi-autonomous, it carries a copy of the control nodes that otherwise would exist in its superiors if those superiors were tightly coupled in an integrated control structure. Because the individual vehicles are physically separated, and may be only occasionally in contact with each other or with their superiors through a low bandwidth and often unreliable communication channel, it is necessary for each vehicle to carry a surrogate chain of command that performs the functions of its superiors in the command chain.

REFERENCES

J. S. Albus, A. M. Meystel, A Reference Model Architecture for Design and Implementation of Intelligent Control in Large and Complex Systems, *International Journal of Intelligent Control and Systems*, Vol.1, No. 1, 1996, pp15-30.

E. D. Dickmanns, et. al., The Seeing Passenger Car 'VaMoRs-P', *International Symposium on Intelligent Vehicles'94*, Paris, October 24-26, 1994